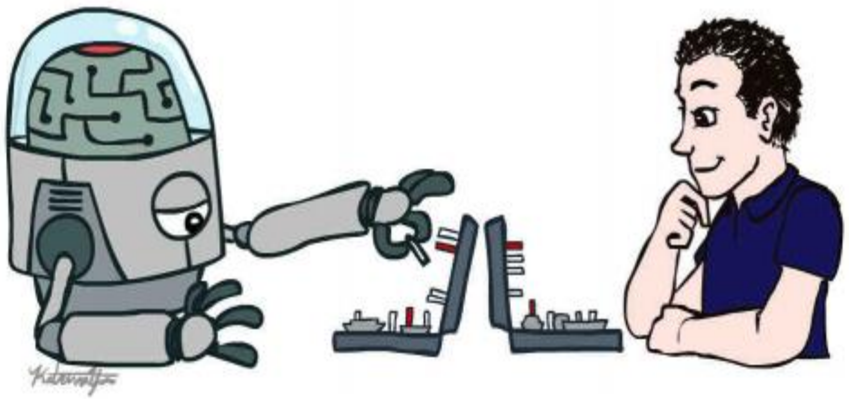
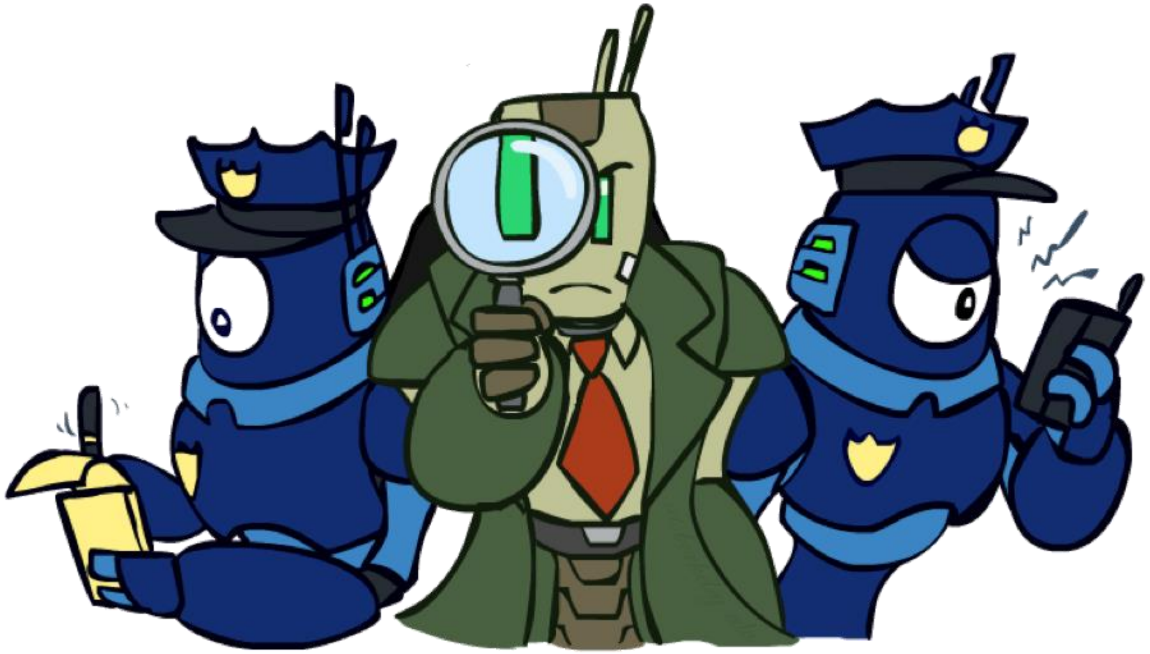
Artificial Intelligence Search



Dr Sherin ElGokhy

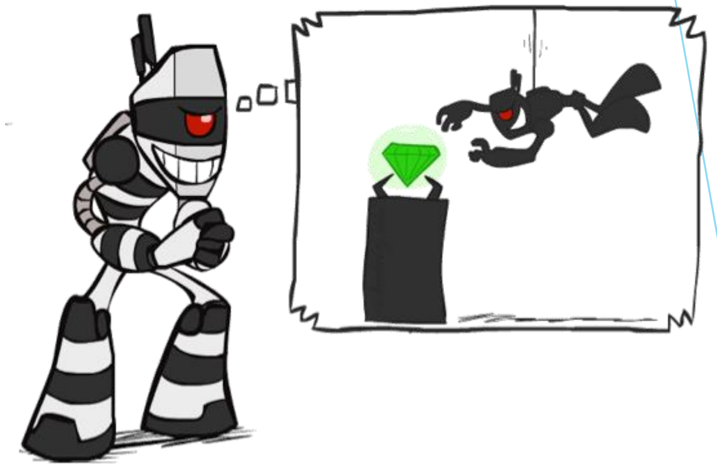
Constraint Satisfaction Problems





What is Search For?

- Assumptions about the world: a single agent, deterministic actions, fully observed state, discrete state space

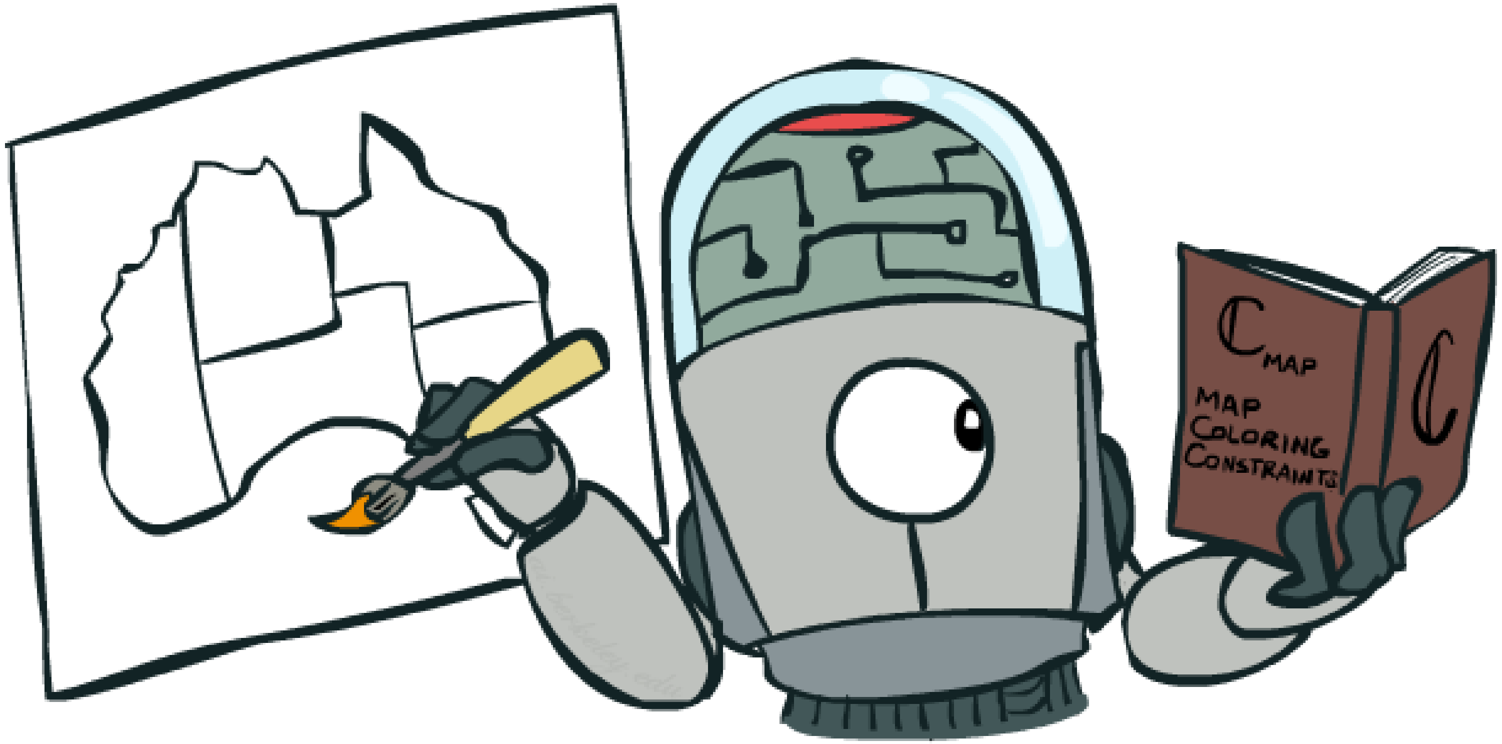


- Planning: sequences of actions
  - The path to the goal is the important thing
  - Paths (Plans) have various costs, depths
  - Heuristics give problem-specific guidance

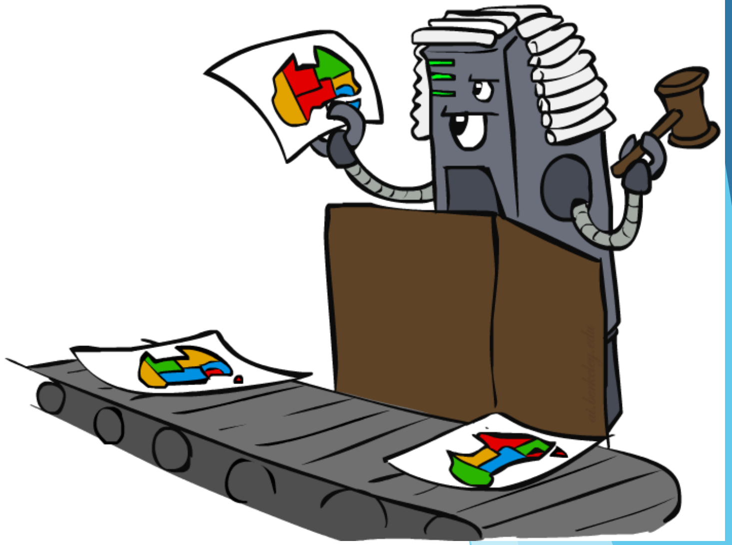
- Identification: assignments to variables
  - The goal itself is important, not the path
  - All paths at the same depth (for some formulations)
  - CSPs are specialized for identification problems



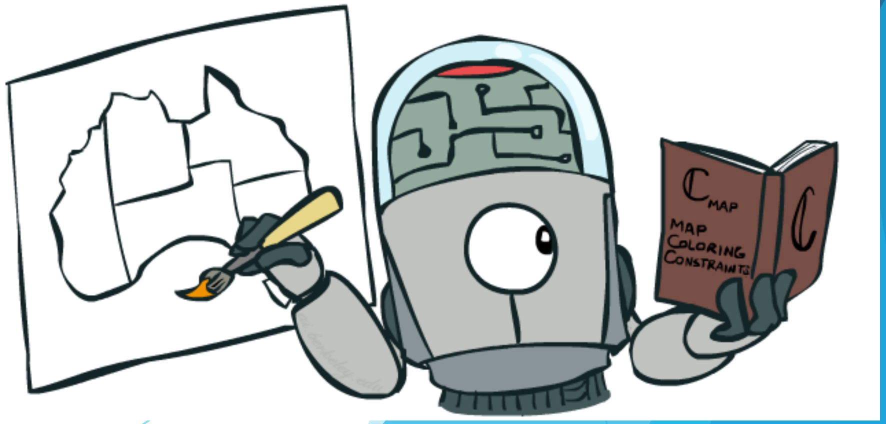
Constraint Satisfaction Problems



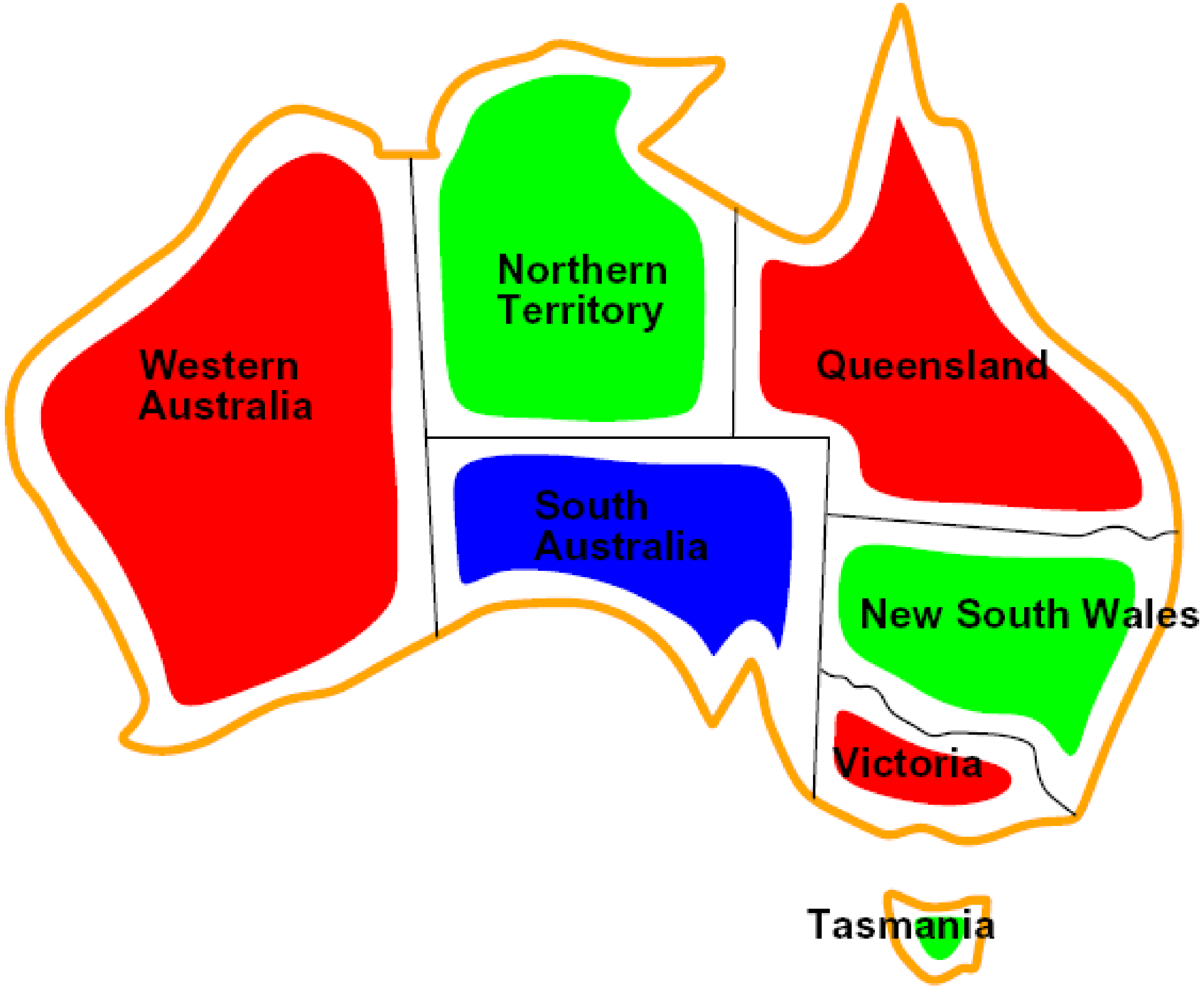
Constraint Satisfaction Problems



- Standard search problems:
  - State is a "black box": arbitrary data structure
  - Goal test can be any function over states that says yes, this is a goal or no, this is not.
  - Successor function can also be anything
- Constraint satisfaction problems (CSPs):
  - A special subset of search problems
  - We make assumptions about what exactly goes into the state
  - State is defined by variables X_i with values from a domain D (sometimes D depends on i)
  - Goal test is a set of constraints specifying allowable combinations of values for subsets of variables
- Simple example of a *formal representation language*
- Allows useful general-purpose algorithms with more power than standard search algorithms



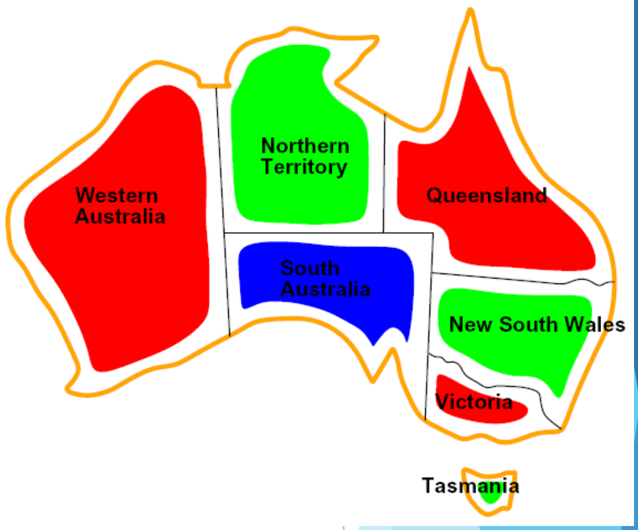
CSP Examples



Example: Map Coloring

- Variables: $WA,\ NT,\ Q,\ NSW,\ V,\ SA,\ T$
- Domains: $D = \{\text{red}, \text{green}, \text{blue}\}$

All the variables have the same domain



- Constraints: adjacent regions must have different colors

  Implicit: $WA \neq NT$

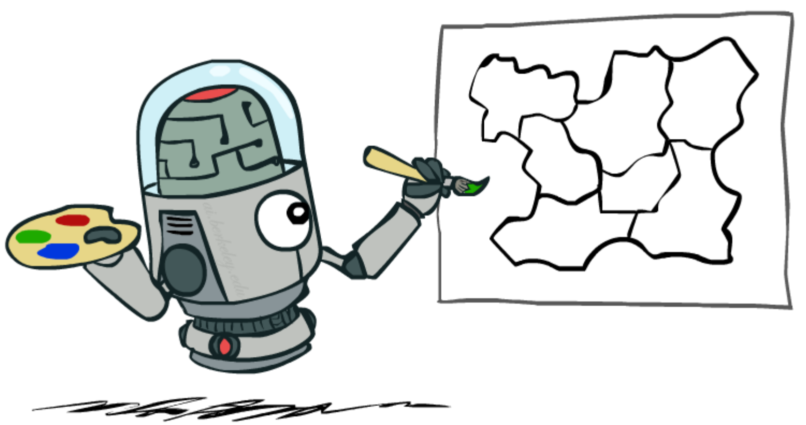
  Means the constraints described implicity by a code snippet

  Explicit: $(WA, NT) \in \{(\text{red}, \text{green}), (\text{red}, \text{blue}), \ldots\}$

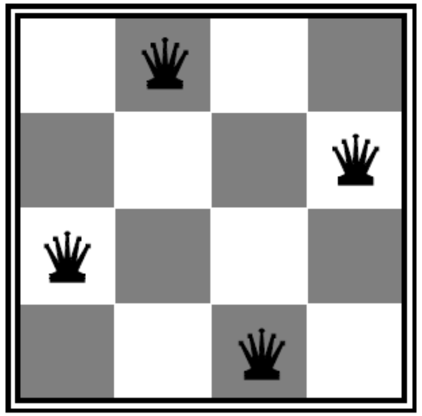
  Means we actually list all of the allowed combinations of assignments

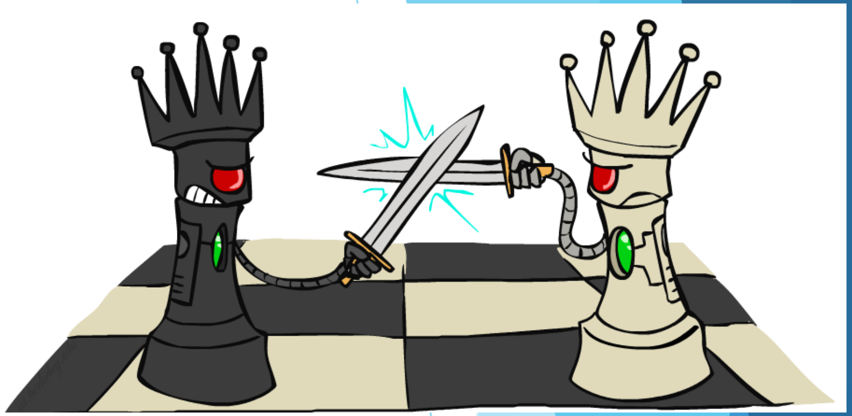
- Solutions are assignments satisfying all constraints, e.g.:

  $\{WA=\text{red},\ NT=\text{green},\ Q=\text{red},\ NSW=\text{green},\ V=\text{red},\ SA=\text{blue},\ T=\text{green}\}$



Example: N-Queens





- Formulation 1:
  - Variables: X_{ij}
  - Domains: $\{0, 1\}$

  The square is either queened or empty

  - Constraints

$$\forall i, j, k \quad (X_{ij}, X_{ik}) \in \{(0,0), (0,1), (1,0)\}$$
$$\forall i, j, k \quad (X_{ij}, X_{kj}) \in \{(0,0), (0,1), (1,0)\}$$
$$\forall i, j, k \quad (X_{ij}, X_{i+k,j+k}) \in \{(0,0), (0,1), (1,0)\}$$
$$\forall i, j, k \quad (X_{ij}, X_{i+k,j-k}) \in \{(0,0), (0,1), (1,0)\}$$

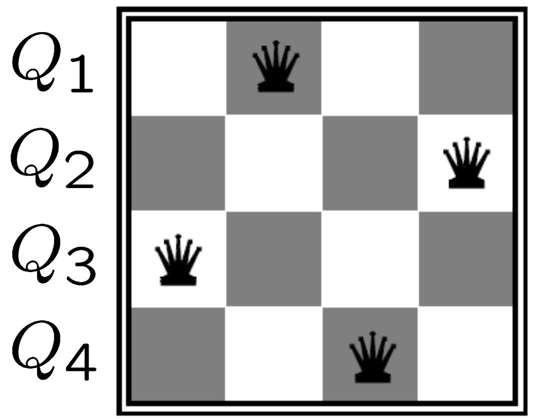
$$\sum_{i,j} X_{ij} = N$$

Example: N-Queens

- Formulation 2:
  - Variables:

    $$Q_k$$
  - Domains:

    $$\{1, 2, 3, \ldots N\}$$
  - Constraints:

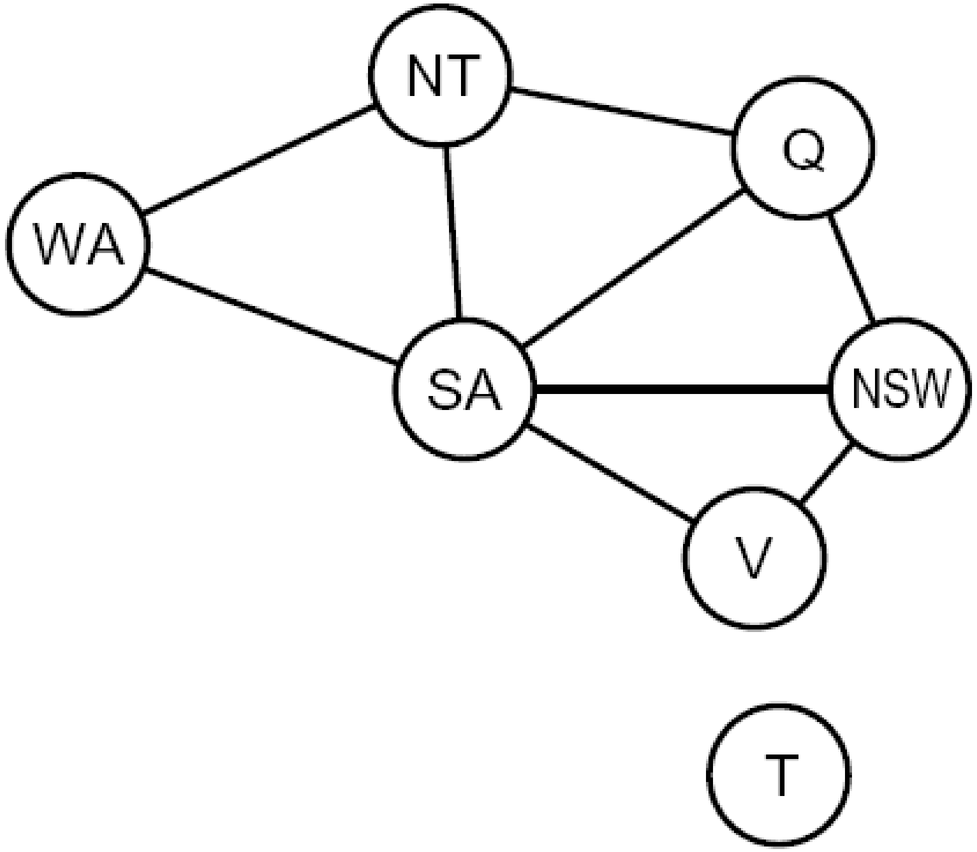


Implicit: $\forall i, j \;\; \text{non-threatening}(Q_i, Q_j)$

Explicit: $(Q_1, Q_2) \in \{(1,3), (1,4), \ldots\}$

$\ldots$

Constraint Graphs

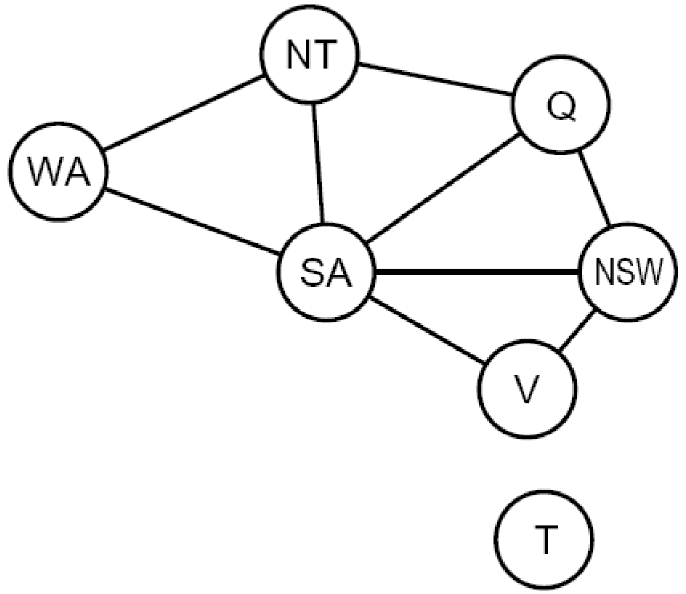


Constraint Graphs

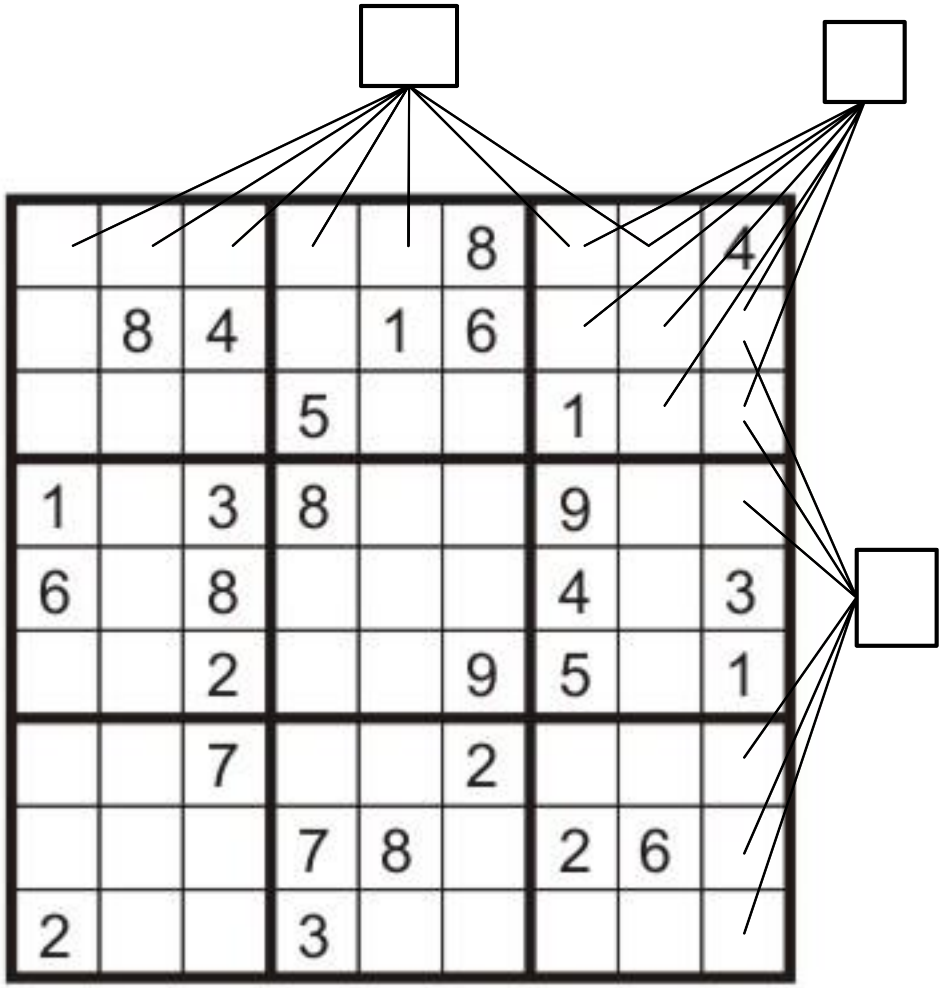
- Binary CSP: each constraint relates (at most) two variables

- Binary constraint graph: nodes are variables, arcs show constraints

- General-purpose CSP algorithms use the graph structure to speed up search. E.g., Tasmania is an independent subproblem!



Example: Sudoku



- Variables:
  - Each (open) square
- Domains:
  - {1,2,…,9}
- Constraints:

9-way alldiff for each column

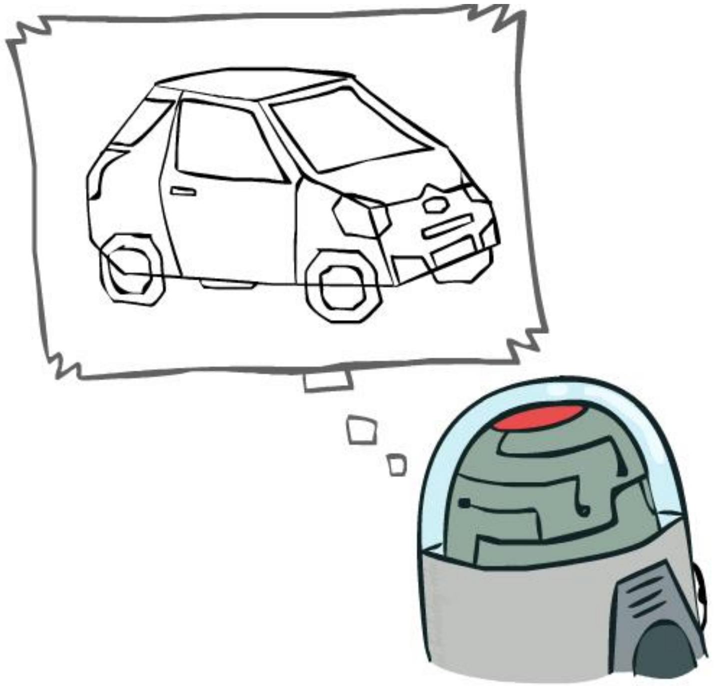
9-way alldiff for each row

9-way alldiff for each region

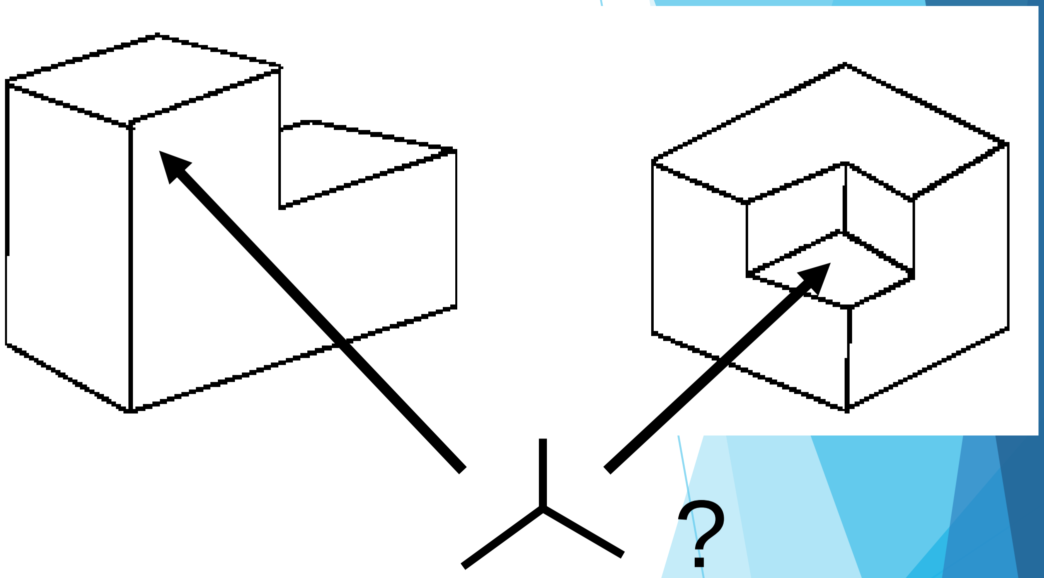
(or can have a bunch of pairwise inequality constraints)

Example: The Waltz Algorithm

- The Waltz algorithm is for interpreting line drawings of solid polyhedra as 3D objects
- An early example of an AI computation posed as a CSP



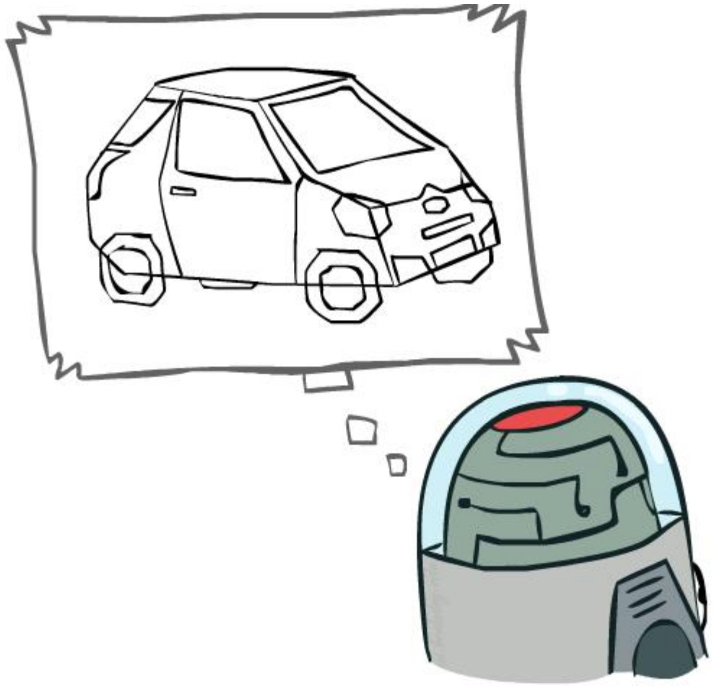




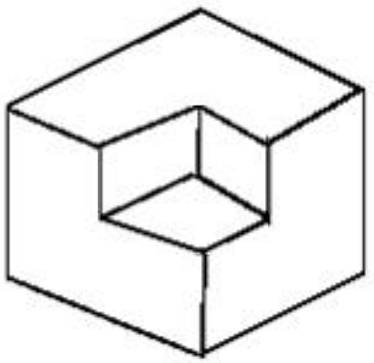
- Approach:
  - Each intersection is a variable
  - Adjacent intersections impose constraints on each other
  - Solutions are physically realizable 3D interpretations

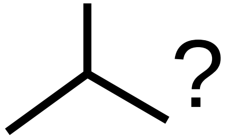
Example: The Waltz Algorithm

- The Waltz algorithm is for interpreting line drawings of solid polyhedra as 3D objects
- An early example of an AI computation posed as a CSP



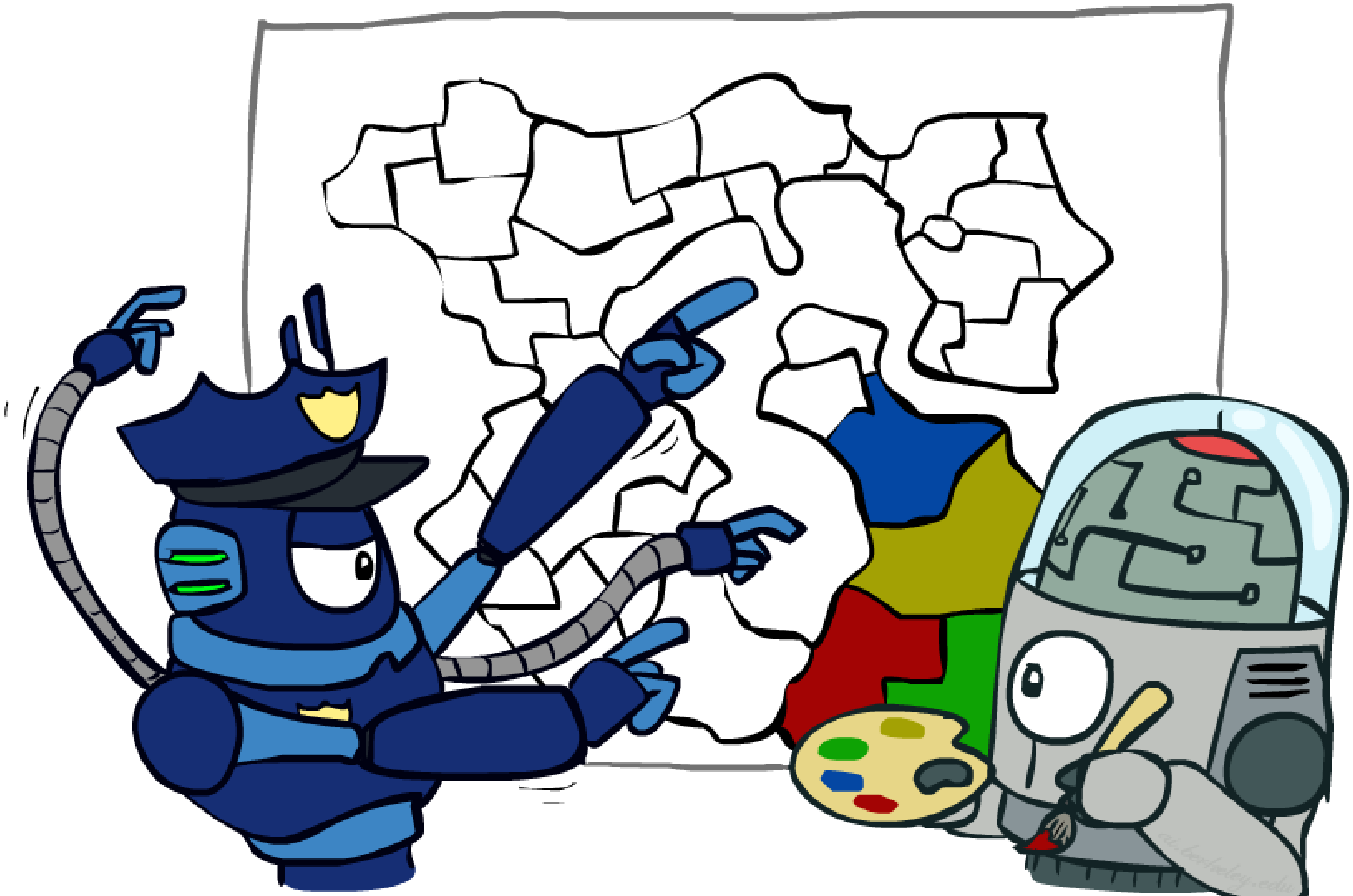






- Approach:
  - Each intersection is a variable
  - Adjacent intersections impose constraints on each other
  - Solutions are physically realizable 3D interpretations

Varieties of CSPs and Constraints



Varieties of CSPs

- Discrete Variables
  - Finite domains
    - Size d means O(dn) complete assignments
    - E.g., Boolean CSPs, including Boolean satisfiability (NP-complete)
  - Infinite domains (integers, strings, etc.)
    - E.g., job scheduling, variables are start/end times for each job
    - Linear constraints solvable, nonlinear undecidable

- Continuous variables
  - E.g., start/end times for Hubble Telescope observations
  - Linear constraints solvable in polynomial time





Varieties of Constraints

- Varieties of Constraints
  - Unary constraints involve a single variable (equivalent reducing domains), e.g.:

$$SA \neq green$$

  - Binary constraints involve pairs of variables, e.g.:

$$SA \neq WA$$

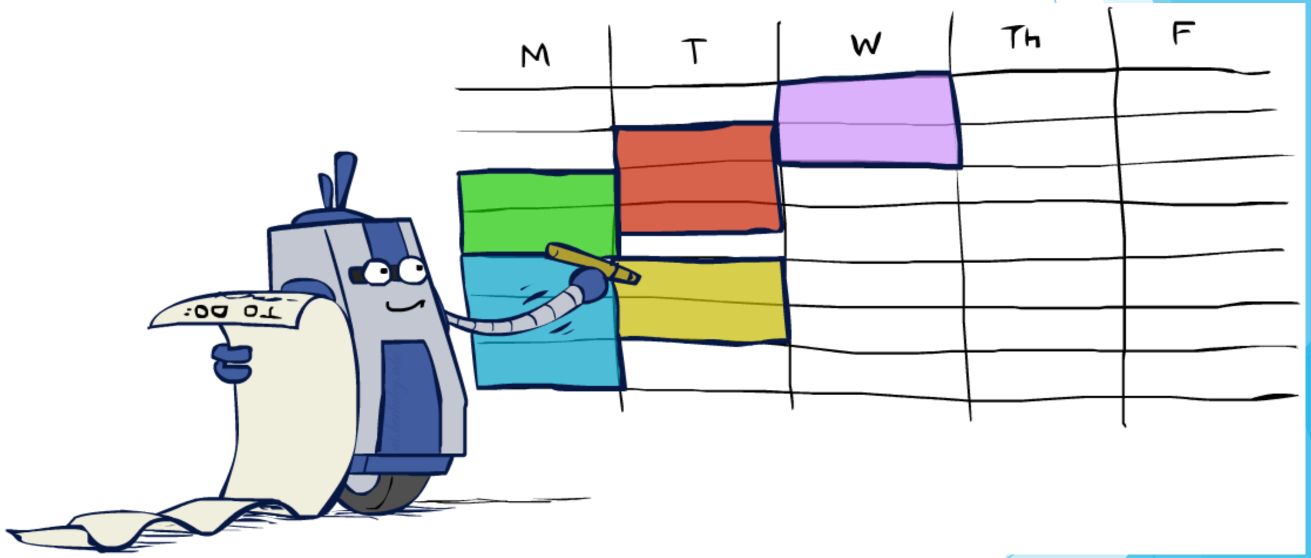
  - Higher-order constraints involve 3 or more variables:

- Preferences (soft constraints):
  - E.g., red is better than green
  - Often representable by a cost for each variable assignment
  - Gives constrained optimization problems



Real-World CSPs

- Assignment problems: e.g., who teaches what class
- Timetabling problems: e.g., which class is offered when and where?
- Hardware configuration
- Transportation scheduling
- Factory scheduling
- Circuit layout
- Fault diagnosis
- … lots more!



- Many real-world problems involve real-valued variables…

Solving CSPs

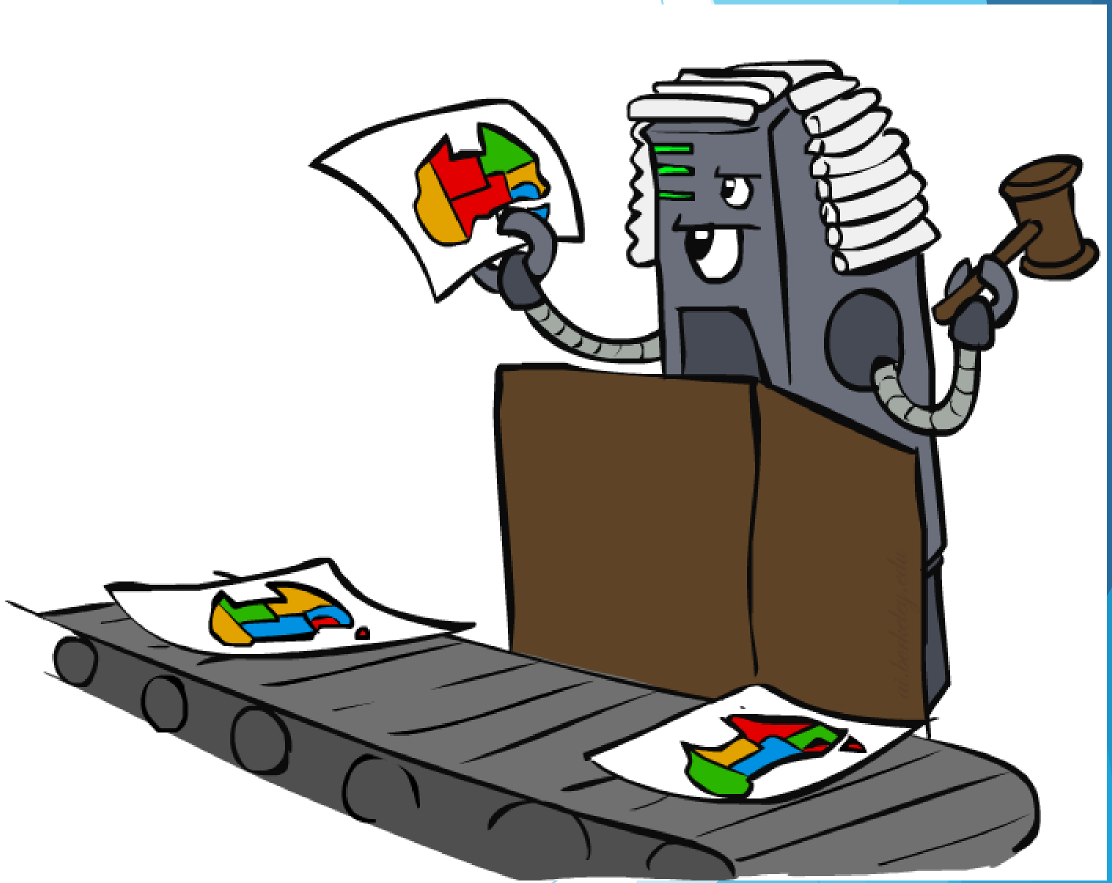


Standard Search Formulation

- Standard search formulation of CSPs

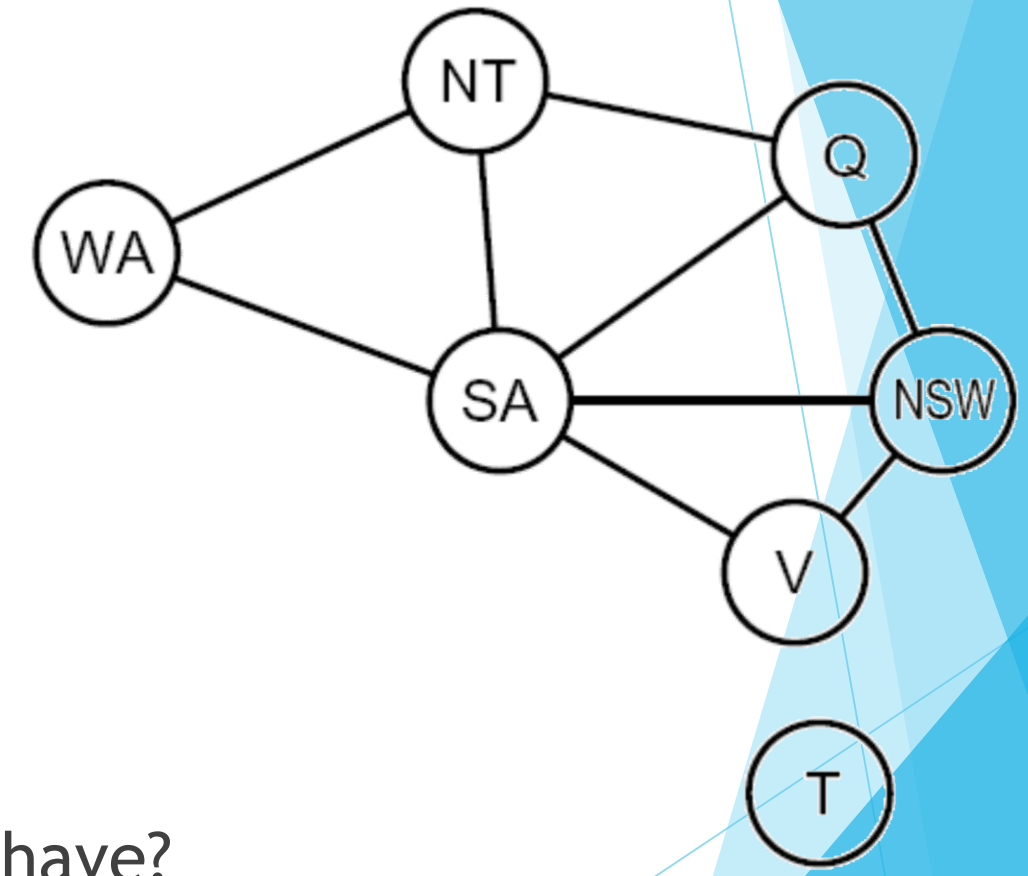
- States defined by the values assigned so far (partial assignments)
  - Initial state: the empty assignment, {}
  - Successor function: assign a value to a single unassigned variable
  - Goal test: the current assignment is complete and satisfies all constraints

- We'll start with the straightforward, then improve it



Search Methods

- What would BFS do?
- What would DFS do?
- What problems does naïve search have?



[Demo: coloring -- dfs]

Thanks